

```

/*****
/*
/*          ALGORITHME          */
/*          */
/*          by GUIRE Elisée Aimé Enoch          */
*****/

```

Algorithme Repertoire()

Variables

Repertoire: ^ListeChaine

Action: entier

choix: entier

// Déclaration des structures

Type

Structure Personne

nom: chaîne

prenom: chaîne

tel: chaîne

mail: chaîne

suivant: ^Personne

Finstruct

Type

Structure ListeChaine

premier: ^Personne

Finstruct

// Fonction pour initialiser la liste

Fonction Initialisation(): ^ListeChaine

Variables locales

nouveau: ^ListeChaine

DÉBUT

Allouer(nouveau)

nouveau->premier <- NIL

Retourner(nouveau)

FIN

// Procédure pour ajouter une personne

Procédure Creer_Enregistrement(liste: ^ListeChaine)

Variables locales

courant: ^ListeChaine

nouveau: ^Personne

DÉBUT

courant <- liste->premier

Afficher("Nom: ")

Lire(nouveau->nom)

Afficher("Prénom: ")

Lire(nouveau->prenom)

Afficher("Téléphone: ")

Lire(nouveau->tel)

Afficher("Mail: ")

Lire(nouveau->mail)

SI courant = NIL ALORS

courant <- nouveau

nouveau->suivant <- NIL

SINON

TANT QUE courant->suivant ≠ NIL

courant <- courant->suivant

FIN TANT QUE

courant->suivant <- nouveau

nouveau->suivant <- NIL

FIN SI

FIN

```

// Afficher le contenu du répertoire
Procédure Afficher_Repertoire(liste: ^ListeChaine)
Variables locales
    courant: ^Personne
DÉBUT
    courant <- liste->premier
    TANT QUE courant ≠ NIL
        Afficher("Nom: ",courant->nom)
        Afficher("Prénom: ",courant->prenom)
        Afficher("Téléphone: ",courant->tel)
        Afficher("Mail: ",courant->mail)
        courant <- courant->suivant
    FIN TANT QUE
FIN

// Procédure pour rechercher une personne par son nom et prénom
Procédure Rechercher_Nom_Prenom(liste: ^ListeChaine)
Variables locales
    nom: chaîne
    prenom: chaîne
    courant: ^Personne
DÉBUT
    courant <- liste->premier
    Afficher("Nom: ")
    Lire(nom)
    Afficher("Prénom: ")
    Lire(prenom)
    TANT QUE courant ≠ NIL
        SI courant->nom = nom ET courant->prenom = prenom
            Afficher("Nom: ",courant->nom)
            Afficher("Prénom: ",courant->prenom)
            Afficher("Téléphone: ",courant->tel)
            Afficher("Mail: ",courant->mail)
        FIN SI
        courant <- courant->suivant
    FIN TANT QUE
FIN

// Procédure pour rechercher une personne par son numéro de téléphone
Procédure Rechercher_Tel(liste: ^ListeChaine)
Variables locales
    tel: chaîne
    courant: ^Personne
DÉBUT
    courant <- liste->premier
    Afficher("Téléphone: ")
    Lire(tel)
    TANT QUE courant ≠ NIL
        SI courant->tel = tel
            Afficher("Nom: ",courant->nom)
            Afficher("Prénom: ",courant->prenom)
            Afficher("Téléphone: ",courant->tel)
            Afficher("Mail: ",courant->mail)
        FIN SI
        courant <- courant->suivant
    FIN TANT QUE
FIN

// Procédure pour rechercher une personne par son adresse mail
Procédure Rechercher_Mail(liste: ^ListeChaine)
Variables locales
    mail: chaîne
    courant: ^Personne

```

```

    next: ^Personne
DÉBUT
    courant <- liste->premier
    Afficher("Mail: ")
    Lire(mail)
    TANT QUE courant ≠ NIL
        SI courant->mail = mail
            Afficher("Nom: ",courant->nom)
            Afficher("Prénom: ",courant->prenom)
            Afficher("Téléphone: ",courant->tel)
            Afficher("Mail: ",courant->mail)
        FIN SI
        courant <- courant->suivant
    FIN TANT QUE
FIN

// Fonction pour supprimer une personne du répertoire
Fonction Retirer(liste: ^ListeChaine): ^ListeChaine
Variables locales
    nom: chaîne
    prenom: chaîne
    courant: ^Personne
DÉBUT
    courant <- liste->premier
    SI courant ≠ NIL
        Afficher("Nom: ")
        Lire(nom)
        Afficher("Prénom: ")
        Lire(prenom)
        SI courant->nom = nom ET courant->prenom = prenom
            liste->premier <- courant->suivant
            Désallouer(courant)
            Retourner(liste)
        FIN SI
    SINON
        next <- courant->suivant
        TANT QUE next ≠ NIL
            SI next->nom = nom ET next->prenom = prenom
                courant->suivant <- next->suivant
                Désallouer(next)
                Retourner(liste)
            FIN SI
            courant <- next
            next <- next->suivant
        FIN TANT QUE
    FIN SINON
FIN SI
Retourner(liste)
FIN

DÉBUT /* Début de l'algorithme */
    Repertoire <- Initialisation()
    Afficher("(1) Ajouter une personne")
    Afficher("(2) Afficher le répertoire")
    Afficher("(3) Faire une recherche")
    Afficher("(4) Retirer une personne")
    Afficher("(5) Terminer")
    Lire(Action)
    SI (Action = 1)
        Creer_Enregistrement(Repertoire)
    FIN SI

    SI (Action = 2)
        Afficher_Repertoire(Repertoire)

```

```
FIN SI

SI (Action = 3)
  Afficher("(1) Par le nom et le prénom")
  Afficher("(2) Par le numéro de téléphone")
  Afficher("(3) Par l'adresse mail")
  Lire(choix)
  SI (choix = 1)
    Rechercher_Nom_Prénom(Repertoire)
  FIN SI
  SI (choix = 2)
    Rechercher_Tel(Repertoire)
  FIN SI
  SI (choix = 3)
    Rechercher_Mail(Repertoire)
  FIN SI
FIN SI

SI (Action = 4)
  Repertoire <- Retirer(Repertoire)
FIN SI

SI (Action = 5)
  Afficher("Aurevoir")
  FIN
FIN SI
FIN /* Fin de l'algorithme */
```